White Paper Report

Report ID: 105436

Application Number: PR-50155-12

Project Director: Karen Cariani (karen_cariani@wgbh.org)

Institution: WGBH Educational Foundation

Reporting Period: 2/1/2012-1/31/2015

Report Due: 4/30/2015

Date Submitted: 2/26/2015

HydraDAM Final Report

Prepared by: Mark Bussey, Data Curation Experts
Karen Cariani, WGBH Media Library and Archives
January 2015

EXECUTIVE SUMMARY

The goals of this project were to test the feasibility of developing a digital asset management system for preservation and access of media materials using open source software and applications that would be easily replicable and implementable by other organizations. We were going to seek feedback specifically from other public media organizations on functionality, documentation, and implementation.

The challenge with managing digital media files stems from the multitude of file formats created from the digital video cameras. The size of the files for finished final programs, the varying codecs, wrappers, file formats, and the sheer volume of content being created that is born digital adds to the complexity of systems needed to manage this content. In addition, long-term preservation and accessibility of these files depends on good storage, easy migration of content and systems. Most current systems for large amounts of media materials are vendor products that are expensive to purchase, maintain and continue to use, and difficult to transition off. With this project we were able to explore the potential of a media asset management system for preservation with open source software.

The project accomplished these goals and more.

- It established a prioritized list of requirements for a media archive.
- It built a proof of concept system based on open source technologies.
- The system both combined existing software and drove the innovation of new technologies.
- The Proof Of Concept was successfully installed at three different sites.
- The functional code / enhancements from the effort have already been incorporated into the Hydra Open Source Software community.
- The effort has helped tap the interest and code contributions from non-public media archives in addressing media archiving needs.
- The project has increased interest from other public media organizations in Open Source solutions
- The project has demonstrated that an Open Source Software -based media archival solution is not only possible, but promising.

OVERVIEW

The goals of this project were to test the feasibility of developing a digital asset management system for preservation and access of media materials using open source software and applications that would be easily replicable and implementable by other organizations. We were going to seek feedback specifically from other public media organizations on functionality and implementation. Our system is called HydraDAM.

We chose to work with the Hydra open source technology stack for several reasons. First, it has been developed and employed by several academic institutions for use in the their digital libraries.

Second, almost all of the functionality needed to manage digital files for digital libraries is necessary for the management of audiovisual materials – ingest, store, discover, retrieve. One organization, the Rock and Roll Hall of Fame, had utilized it building a system for access to audio visual materials. Indiana University and Northwestern were using it to develop an access system for their media materials. But when we had started nobody had utilized it for media preservation. For preservation there is a lot of technical information that needs to be captured. In addition because the files are big, they are generally stored outside the data repository, so their location needs to be noted, and workflows and coding need to be written to accommodate that.

Third, we also chose the Hydra stack because of the strength of the Hydra community. It is a strong open community that has been growing steadily, for the last four years. The institutions adopting Hydra have an established reputation for programmatic commitments and sustainable resources to maintain their technology. Many of its current members have media in their collection that requires preservation, providing a natural pool of potential contributors and adopters. And as more organizations adopt Hydra as a framework, functionality originally developed in other contexts can be beneficially applied to media preservation solutions.

And fourth, the Hydra community has a strong sense of openness and inclusion. The community is very committed to sharing and supporting across institutions. The public media sector aligns well with the Hydra community in terms of resources, approach and culture, providing a promising venue for inclusion and future collaboration. The institutions adopting Hydra have an established history of using sustainable resources to keep their technology growing. Being a member of this community means you can share in the development and help steer the direction for new solutions.

So what is Hydra? It is a powerful and flexible framework designed to accommodate a variety of digital asset needs. A robust repository (the “body”) provides back-end management and preservation capabilities. Customizable applications and

workflows (the “heads”) allow tailored ingest and access depending on the content type, site and user needs. There are collaboratively built gems and “solution bundles” that can be leveraged or adapted and modified to suit local needs. The code is open and available through an Apache license.

For more info see: http://projecthydra.org

A very compelling reason to adopt Hydra’s technologies are that they take advantage of the benefits of the underlying Fedora Commons digital repository. Fedora (Flexible Extensible Digital Object Repository Architecture) provides digital object middleware that can track and manage rich relationships among multiple files able to form complex digital objects. This allows for a complex web of networked data, as opposed to the rigid, hierarchical structure common to databases. Fedora also defines a set of abstractions for expressing digital objects, asserting relationships among digital objects, and linking "behaviors" (i.e., services, attributes) to digital objects. By managing objects as a network of files, and managing services on those objects with a preservation mindset, the Fedora repository software enables long-term access to digital resources. Rich media is complicated and the data needed to manage media is complicated and varied – technical data, provenance data, descriptive data, copyright information, access permissions, related documents like transcripts, contracts, releases, preservation, etc. Therefore a repository like Fedora is a natural fit for managing these digital objects, allowing different views of data for different aspects of the media.

For more info see: http://fedorarepository.org/about

There are other Fedora-based solutions like Islandora, but Hydra’s strength is its community. At a time when we are all asked to do more with less, and faster, working in a community where everyone has a shared purpose and goals, makes that easier. We all need systems to do basically the same thing in managing our digital objects. Why not leverage that and work together?

There is a community of developers and adopters extending and enhancing the core framework. The community is friendly and welcoming. There is training thanks to Data Curation Experts Hydra Camp (an intensive introduction and training for developers on Hydra code) and vendor support – you can hire them to help develop, train, and support your staff. And they are also very much a part of the sharing community committed to moving the overall technology forward. We worked closely with Digital Curation Experts on this project.

PROJECT CONTEXT

HydraDAM is a digital asset management system aimed at the needs of public broadcast institutions and other institutions seeking to manage heterogeneous

collections of video and audio archival materials, including final-form production media, unedited born-digital production assets, digitized proxies of analog media, along with catalog and archival records for external (non-digital) final productions and original media assets.

This section summaries the project's context, including practical and technical issues facing the public media community that served as drivers for the initial project scope. A discussion follows describing how the project responded to each specific deliverable area in the project grant. In addition to the specific grant deliverables, the report then describes a number of additional features developed in order to provide a coherent and fully functioning system that could be packaged and implemented by interested institutions. A further list of enhancements are then documented which have not been implement at this time, but were identified during the project as highly desirable features which would support additional use and adoption. Finally, the report concludes with a review of the project within the larger public media and open source communities.

Initial development has been funded by a grant from the National Endowment of the Humanities. Grant scope included the development of a proof-of-concept open source application and feasibility and usability evaluation of that application by various public broadcasters and potentially other cultural institutions.

As the grant recipient, we developed specific feature requirements, directed the overall development process and provided subject matter expertise on technical, archival, and administrative requirements. WGBH partnered with Data Curation Experts (DCE), a small software development and consulting firm specializing in digital repository solutions, to undertake the actual development of the application. During the course of the project, WGBH and DCE also participated in a number of activities aiming to encourage sufficient knowledge transfer to WGBH staff to ensure the long-term sustainability of and media management strategy based on the new application.

In order to ensure broader suitability of the application, WGBH engaged additional media partners to evaluate the solution at different points of development. WNYC was brought on board to evaluate a beta release of the application and provide both functional and implementation feedback. WNYC provided valuable feedback that led to modification in application functionality along with installation and management improvements. South Carolina Educational Television (SCETV) was brought in at a later phase of the project to evaluate a simplified and more fully documented version of the installation process, along with a more mature version of the application. The final feedback from SCETV was positive.

Original Proposal Project Goals

- To develop a Fedora-based integrated, sustainable, open source digital asset management solution for media that qualifies as a Trustworthy Digital Repository, with an open modular architecture based on the OAIS reference

model, and serves the preservation and access needs of media producers, archives, and researchers.
- To develop a content agnostic system that is flexible enough to evolve over time as media technology changes, but is sufficiently opinionated to be easy to use and fulfill the needs of the community.
- To develop tools and documentation that allows the system to be easily implemented on common off-the-shelf hardware while allowing for scalability to petabytes of material.
- To assemble an end-to-end digital asset management system that provides solutions and services to support the needs of the public media community, while building on ongoing work in different communities of practice,.

In order to make the most of available funding, scope was narrowly focused: user interface, discovery, and distribution functionality were eliminated in favor of key features required to support archival activities. Key archival features included ingest and transcoding of audio and video media files, and metadata management using PBCore as the standard for descriptive metadata. In addition to this functionality, the scope also included initial exploration of operational issues including integrations with Hierarchical Storage Management (HSM) systems, and deployment strategies for the application.

Project Deliverables

The project's originally proposed deliverables were to:

- Develop and publish a list of key requirements for a digital media system for preservation (per TRAC – Trusted Repositories Audit & Certification) including different storage modules for different storage solutions, including large scale Hierarchical Storage Management (HSM), basic file system storage, and cloud storage for large and small files.
- Develop specifications and best practices for implementing a Fedora repository for media materials, especially around the needs of large file storage and delivery, storage back-ends, and workflows.
- Integrate applications for the management and assessment of file storage in complex environments including HSM and cloud-based storage, in order to ensure the digital media files are valid, complete, and accessible after ingest.
- Develop documentation needed for implementation of the digital repository bundle.
- Develop written documentation needed for easy implementation and local customization of the Hydra "head".
- Create specifications and recommendations for ingest models for at least 3 common file formats in the media industry, especially around preservation and access needs, including media transcoding technical metadata extraction, and the creation of access copies and thumbnails.

- Develop ingest workflows and tools that allow bulk ingest and ease the process of reconnecting media assets with existing metadata records.
- Develop basic access and management interfaces within the Hydra framework to support discovery and metadata cataloging/enhancement.
- Develop and prototype connections between the digital repository stack and existing publishing outlets.

Design Considerations

Because of the decision to remove user interface concerns from scope, DCE recommended using an existing Hydra application as a starting point which would provide a limited user interface for testing and exploratory purposes. We selected Sufia, a gem encapsulating functionality developed by Pennsylvania State University in their ScholarSphere application. ScholarSphere's focus on managing arbitrary file assets fit well with HydraDAM's focus on media files. Inheriting ScholarSphere's user interface for deposit and metadata management with minimal investment allowed us to focus the majority of our efforts on the following technical concerns: ingest, transcoding, PBCore compatibility, HSM integration, and deployment concerns.

PROJECT OUTCOMES BY DELIVERABLE

Requirements for digital media preservation systems

Develop and publish a list of key (must have) requirements for a digital media system for preservation (per TRAC) including different storage modules for different storage solutions, including large scale Hierarchical Storage Management (HSM), basic file system storage, and cloud storage for large and small files.

We developed the requirements document looking at the TRAC (Trustworthy Repository Audit Checklist, a precursor to the TDR specification) guidelines to help us with specific functionality requirements. We developed a requirements list and received feedback from our advisors, which we took into consideration, especially when it came to prioritizing features for implementation. In the end WGBH committed additional, internal (non-grant-based) funds to complete some of the features that were deemed important to have. See page 23 for final system functional requirements.

The system is capable of integrating with any storage system that presents itself as a file system. The coding will be specific and unique for each storage system. We coded for the WGBH system, and other sites can use this pattern to integrate with other systems.

Best practice guidelines

Develop specifications and best practices for implementing a Fedora repository for media materials, especially around the needs of large file storage and delivery, storage back-ends, and workflows.

The project uncovered a number of gaps in the current release of Fedora related to the management of large files. For the purposes of this project, large files were generally considered to be individual files larger than 1 Gigabyte and collections of large files where the majority of files were over 0.5 Gigabytes. In addition to consuming significant storage resources, files and collections of this size presented transfer, copy, and rendering challenges even on well-connected, high-speed internal networks.

The table below gives representative times for the transfer of 1 Gigabyte of data. Performance may vary depending upon specific circumstances and infrastructure configuration. The key point however, is that even the fastest internal operations occur with a noticeable delay from a user perspective.

Table 1 - Representative times to transfer 1 Gigabyte

Connection Type	**Transfer Rate in Megabytes**	**Average copy time 1GB**
Internal Server Copies		
Internal server copy fast server Solid-State Drive	~95MB/s	10s
Local Copy – external Fire wire drive (theoretical)	~80MB/s	13s
Local Copy – external USB 2.0 drive (theoretical)	~40MB/s	26s
Server to Server copy – same datacenter		
Gigabit Ethernet optimal	~100MB/s	10s
Gigabit Ethernet typical	~35MB/s	30s
Offsite connections		
Offsite connection – High speed = 6 megabit	~6MB/s	28m 26s
Offsite connection – Residential = 1.5 megabit	~1.5MB/s	1h 53m

The system has been designed with two goals to mediate this issue:

1. Minimize the number of times a file is moved
2. Perform as much processing as possible via asynchronous background jobs

The latest version of Fedora available for this project, version 3.7.1,as well as all earlier 3.x releases, require that content be located on the local server running Fedora before being ingested into the repository. Ingest involves classifying the data and copying into the fedora object store. Therefore, typical ingest workflows require at least one duplication cycle of content from an offsite location to the server, and another round of copying the content locally from a temporary storage location into the fedora repository. The application handles the second copy process in the

background and notifies the user when the copy procedure and other processing are complete.

There are no technical solutions to the delay involved by the offsite copy procedure; however, in certain cases it may be more expedient to transfer files from a content producer's local system to a portable USB or FireWire hard drive and then send that drive to the datacenter hosting the application to be connected directly to the server. This process can reduce the time that systems on either end of the transfer are blocked in the copy process. There are, however, logistical, staffing, and security impacts from this process that not all organizations will be resourced to handle.

We have therefore assumed that typical use cases in order of priority will be:

1. Browser-based upload of content over a remote web connection
2. FTP file transfer of files from a remote public or private internet connection
3. Transfer of files via external media such as a portable drives or optical disk

From the application's standpoint, both FTP and external media transfers look the same: a process outside the system copies the content to a temporary local storage location, and then a browser-based user interface allows the administrator or archivist to select the locally stored files for ingest and processing. For browser based upload, file selection on the user's computer and transfer are integrated into a single user action.

The upcoming Fedora 4.0 release may eliminate the need for the local copying step. The opportunities for this functionality are discussed in the "Next Steps" section.

Storage system integration

> *Integrate applications for the management and assessment of file storage in complex environments including HSM and cloud-based storage, in order to ensure the digital media files are valid, complete, and accessible after ingest.*

The HydraDAM application supports any storage system that can be mounted to the local server and appears as part of the native file hierarchy. The system is therefore compatible with traditional spinning disk storage, SSD, RAID, SAN, and HSM storage infrastructures. The initial implementation has provided limited HSM integration that allows the system to indicate to users when selected assets are offline and notify users, via the user dashboard, when requested files have been restored online.

Fixity checking and other auditing

> *Integrate applications for the management and assessment of file storage in complex environments (include HSM and cloud-based storage), in order to ensure the digital media files are valid, complete, and accessible after ingest.*

The HydraDAM application leverages the Fedora repository's native audit and fixity functionality. For performance reasons, Fedora's default configuration disables both of these features; however, these features are easily enabled by following the relevant Fedora documentation. Because of the potential performance and preservation impacts of excessive fixity checking, those areas have currently been relegated to local practice at the implementing institution. A logical enhancement of current functionality would be to expose and manage the underlying Fedora capabilities directly through the HydraDAM interface, as discussed in the Next Steps section.

Implementation and support documentation

Develop documentation needed for implementation of the digital repository bundle.

Develop written documentation needed for easy implementation and local customization of the Hydra "head".

Data Curation Experts developed draft implementation documentation and then solicited feedback during various rounds of testing by WGBH, WNYC, SCETV, and other members of the Hydra community. We anticipate that this documentation will grow and evolve as the application matures; however, we feel that given the broad review and multiple tests of the documentation that is it complete and robust as it stands. The installation documentation can be found in the wiki associated with the application: https://github.com/curationexperts/hydradam/wiki/Production-Installation:-Overview. This type of installation documentation had not previously been available within the broader HydraDAM community and has benefitted a number of institutions beyond just those implementing HydraDAM.

In addition to the implementation documentation, preliminary operation and customization documentation has also been developed. It has been initially difficult to identify what gaps exist in existing Hydra operational documentation elsewhere. Therefore, HydraDAM documentation has been provided in a wiki-based format to allow easy modification, clarification, and extension of the documentation as additional needs are identified by repository users and systems support staff. At present, the documentation consists of the Production Installation Guide, a Developer Setup Guide, an Operations Guide, and a Troubleshooting Guide in FAQ format. A homepage for all of the existing guides can be found at: https://github.com/curationexperts/hydradam/wiki.

Ingest, characterization, and transcoding

Create specifications and recommendations for ingest models for at least 3-4 common file formats in the media industry, especially around preservation and access needs, including media trans-code, technical metadata extraction, and the creation of access copies and thumbnails.

Based on evaluations of their own holdings, and feedback from the advisors and partners, WGBH elected to focus ingest development and testing on three video file formats and two audio formats:

- MOV – Apple's QuickTime format files, frequently produced as output from various production and editing applications
- MPEG-4 – ISO standard video and multimedia file format
- AVI – common multimedia/video file format used by many consumer and professional digital video cameras
- WAV – uncompressed digital audio files
- MP3 – common internet format for compressed digital audio

The application will recognize each file type, extract relevant technical metadata, and render web-friendly proxies or derivatives of the original media to be played back in users' browser.

By selecting the Sufia repository gem as a starting point, HydraDAM inherited a variety of Sufia features useful for ingest and transcoding of media files:

- Background characterization and rendering of derivatives. Once files have been uploaded, file characterization (determining what type of content the file contains), technical metadata extraction (format, bit-rate, run-time, etc.), and rendering (producing lower-resolution versions to play back over low-bandwidth internet connections) occur in the background. The user is free to proceed to other work such as uploading additional files and/or adding descriptive metadata to files that have been previously uploaded.
- File characterization and basic technical metadata extraction using the File Information Tool Set (FITS). The software was initially created by the Harvard University Library Office for Information Systems and now released and publicly maintained under GNU LPGL license.
- The FFMPEG framework allows additional technical metadata extraction for media files and provides the rendering libraries for media files. The use of FFMPEG transcoding libraries allows for the future extension of HydraDAM to handle any number of the FFMPEG supported media formats. (for a complete list of currently supported media formats, see http://www.ffmpeg.org/general.html)
- Pre-existing support for a variety of document and image file formats including common office formats such as Word (.doc), Excel (.xls), Powerpoint (.ppt), open source equivalents such as Open Office and Libre Office, Adobe Portable Document Format (PDF), and most common image formats supported by the ImageMagick library including TIFF, JPEG, and JPEG2000.

This choice allowed us to focus on implementing the necessary support for the selected multimedia file formats. The Sufia gem did not previously support these formats. The decision benefitted the HydraDAM application by providing support for a variety of additional file formats, enabling the repository to manage not only audio and video content, but also supporting materials such as scripts, production

documentation, and photographic stills associated with various productions. The decision benefitted the larger Hydra community by extending the capabilities of Sufia-based repositories, enabling them to better manage audio and video content alongside documents and images.

Bulk ingest

Develop ingest workflows and tools that allow bulk ingest and ease the process of re-connecting media assets with existing metadata records.

The application implements functionality to support the bulk ingest of metadata exported from WGBH's existing FileMaker based metadata record system. In conjunction with bulk metadata upload, the system provides a metadata to file matching function which allows the administrator or archivist to streamline the process of matching metadata records to their corresponding media asset file or files. Media files may be uploaded through one of three methods:

1. Directly via the browser-based web interface:
   This method is suited to ad-hoc upload of smaller file collections.
2. Via independent FTP upload with selection:
   This method is suited to ad-hoc and routine upload of larger file collections and/or individual larger files.
3. Via connection of external portable hard drives:
   This method is particularly suited to workflows designed to capture the assets for an entire production as the metadata matching utility can be used to match metadata to files based on their location in the drive's directory tree.

Collectively, these capabilities are designed to provide WGBH a viable solution for mass migration of their content into the HydraDAM repository application from production units.

The current bulk ingest capabilities require imported metadata to conform to the format used by WGBH's local FileMaker database. Other Hydra implementations are developing more generalized solutions for bulk metadata and file management. As outlined in the Next Steps section, a desirable future enhancement will be to integrate these newly developed batch capabilities into the HydraDAM application.

Discovery and management Interfaces

Develop basic access and management interfaces within the Hydra framework to support discovery and metadata cataloging/enhancement.

The Hydra repository framework provides default integration with Blacklight's search and discovery tools. Blacklight, a core piece of the Hydra stack, provides straightforward search functionality along with faceted browsing of repository content stored in Fedora. Although Blacklight supports nearly limitless

customization, HydraDAM makes use of the default configuration to meet or exceed all of the initial search requirements defined for the application.

By default, the Sufia gem only supports simple Dublin Core elements for metadata. A significant portion of the development effort was allocated to implementing a larger data model that supported the management of PBCore based metadata. HydraDAM supports a rich description of digital assets using PBCore based metadata terms that are edited via a simple to use browser-based web form. Metadata for any object in the repository can also be viewed and exported in PBCore XML format for interoperability with other media systems supporting the PBCore standard.

Interoperability with other systems

Develop and prototype connections between the digital repository stack and existing publishing outlets.

HydraDAM incorporates a number of underlying technologies to allow interoperability with other systems:

- Fedora directly supports export to OAI-PMH compliant applications for the exchange of descriptive and technical metadata.
- The underlying system stores data in RDF format (specifically, RDF triple statements stored as plain text) rather than PBCore XML. This positions HydraDAM to be effectively extended to support a variety of linked open data applications in the future.
- The system can also display and export stored metadata in PBCore XML format for systems that implement this standard such as typically other public-broadcast focused applications and tools

Since development of HydraDAM began, a newer release of Blacklight has implemented features to support the inclusion of machine-readable microdata in search and browse results and item level views. Exposing this capability within HydraDAM would be a logical next step to support system interoperability with a broad variety of other systems and tools. This requires a classification of A/V and PBCore terms into Schema.org, which is an undertaking.

ADDITIONAL FEATURES

In addition to the functionality outlined in the core requirements, the system implements a variety of additional features to support ease of use and enhance system functionality.

Item-level access control

The Hydra framework supports flexible, item-level access controls, that were initially developed for use in-Pennsylvania State University's ScholarSphere system. The system supports four tiers of access for any object:

- **PUBLIC:** anyone accessing the system can search for and view the item without needing to log-in.
- **INSTITUTIONAL ONLY:** only logged in users with a valid institutional account can search for and view the item.
- **RESTRICTED:** only specifically identified users and groups have access to search and view the item.
- **PRIVATE:** only the owner of the item (and system administrators) can search for and view the item.

Because Hydra implements *'gated discovery'*, users who do not have rights to view an item will see the item in search results when they perform searches across repository contents, but will not be able to view them individually.
In addition to access control, HydraDAM can also be used to store and track the usage rights for an item. This consists of a manually assigned field that can specify the particular licensing or copyright status of the item.

Version Tracking

HydraDAM implements versioning for file assets and tracks whenever a new version of the file associated with an item is uploaded. For metadata tracking, HydraDAM inherits the native version tracking capabilities of the underlying Fedora repository. In the default installation, metadata version tracking is not enabled, but can easily be enabled by the implementing institution.

User-level Dashboards

Each user of the system has access to an individual dashboard providing access to their uploaded content as well as notifications and other status information. Via the dashboard or the general search tool, users can access their uploaded content to make changes to descriptive metadata, update access rights, or upload new versions of the attached file(s).

User notifications

The dashboard provides a means to display status updates about ingest progress, derivative generation, and other background tasks. This allows the system to handle time-consuming processes such asynchronously while the user continues to use the system for other tasks.

Administrator Dashboards

In addition to user-level content dashboards, the system provides two administrative dashboards. One dashboard allows administrators to review the status of background jobs and processes in order to ensure the overall functionality of the system. Another administrative dashboard allows administrators to manage

user accounts, assign administrative rights, and manage user's upload and temporary storage directory locations.

While none of the features listed above were explicitly included in the grant scope, these functions were determined to be necessary to provide a set of coherent functions that could be used in real-world scenarios.

DEPLOYMENT AND PARTNER TESTING

A core goal of the initial project was to identify the requirements for easy deployment of the system to public media partners that may have limited technical resources to manage the system.

- Hydra has traditionally been a developer and tech-centric community and we were treading relatively new ground in attempting to create a repeatable install process for the application.
- As described above, the project has developed a thorough and well-tested set of manual installation instructions for deployment of the application. In addition to sharing these instructions with partners in the media project, DCE has been able to share these instructions and gather feedback from other members of the Hydra community to the benefit of both the other community members of the community and the HydraDAM project.
- We have also tested providing a virtual machine (VM) implementation of the system. This provides a simple means for potential adopters to "kick-the-tires" and explore the capabilities of the system; however, the VM solution is not currently performant enough to support large production implementations. Additional work will be required if a VM version of the application suitable for production deployment is desired.
- Partner testing proved to be an unexpectedly challenging and time consuming portion of the project, but also provided some of the best feedback relating to practical issues impacting adoption of the HydraDAM application. We are very hopeful that this feedback will be incorporated into a further phase of development that would help make the application easier for smaller organizations to adopt out-of-the box with minimal customization.

As part of the deployment and partner testing, installation documentation was produced which filled a previously existing gap in the Hydra ecosystem. In addition to supporting the HydraDAM installation process, the documentation was developed to support more general Hydra implementations. Therefore, in addition to review by WNYC and SCETv, the installation documentation produced for the project has been used and reviewed by a number of other Hydra adopters including West Virginia University, the University of Alberta, Princeton University and others.

NEXT STEPS

Enhanced Large File support

A primary challenge faced by any video repository system is the overhead involved in moving and storing the sheer number of bits involved. HydraDAM attempts to minimize this overhead by minimizing operations against the original archival digital video. Due to limitations of the current Fedora 3.x repository, the original video file must be uploaded from the remote source to a temporary storage location on the HydraDAM server and then copied to its final storage location. The upcoming 4.0 release of the Fedora repository software addresses this issue by supporting upload of the file directly to its final destination, eliminating one time consuming copy operation.

Additionally, Data Curation Experts has begun investigating various background replication services which allow the end-user to copy the archival video to a storage location on their local machine which is then replicated to the server in the background as bandwidth is available. The total time for offsite connections listed in Table 1 above is still in effect; however, implementing a background replication process can make the process more transparent to system users and provides automatic recovery from temporary network dropouts and other transient connection errors.

Enhanced HSM support for selected systems

The initial pilot provided generic hooks for integration with Hierarchical Storage Management (HSM) systems; however, it became clear early in the project that tighter integration with a specific HSM systems was necessary to provide the optimal user experience. Because the underlying Fedora 3.x repository does not provide the necessary integration points, it is not currently possible to implement a closer integration. Fedora is not capable of recalling files that have been migrated to tape and if it attempts to use a file that has been migrated off to tape, it will fail. HydraDAM implemented a feature that brokers Fedora's access to a file based on its status with the HSM. If the file has been migrated, access is blocked and the user is made aware. The user can make a retrieval request and have the file restored to disk. Once this has occurred, the user is notified and HydraDAM will allow Fedora to access the file. Here again, we anticipate features in the upcoming Fedora 4.0 release will facilitate an HSM-aware repository that can provide appropriate notification and storage hooks for the video repository.

Integrated audit and fixity reporting within the application

The current system tracks and logs user access and file activity; however, no unified reporting tools exist. In the current system, ad-hoc reporting requires developer access with relatively detailed knowledge of the system. Once the system has been in use for 6-12 months by a number of pilot institutions, we believe that system owners and administrators will have enough familiarity with the system to

formulate a detailed set of requirements for a core set of reports that could be collectively developed and provided upon installation.

User Interface and User Experience refinement

The initial grant scope focused exclusively on back-end repository functionality and specifically excluded user interface components. In order to provide a usable system for end-user testing, the decision was made to leverage the existing Sufia user interface. While Sufia's interface was an exceptionally effective and low-cost solution for a user interfacet, it was not optimized for audio and video management activities. Therefore, there was definite room to refine the user interface and provide a user experience more tailored to media-specific workflows.

Operational Documentation

As part of the initial pilot, a preliminary framework and templates were generated to support the development of operational and troubleshooting documentation. A set of basic documents have been created based on issues and questions that arose during piloting at WGBH, WNYC, and SCETv. The documentation has been created within a WIKI in order to facilitate easy maintenance and augmentation as the project evolves.

Enhanced Search

The system provides basic search and faceting capabilities provided by the Blacklight search interface; however, preliminary use has suggested a number of additional search capabilities that would be useful, but would have required resources beyond the original grant scope to implement. As a first pass, Data Curation Experts has identified the integration of the Blacklight Advanced Search gem into HydraDAM as a high priority enhancement to the system's overall search capabilities. Additionally, more complex enhancements such as the integration of full-text searches of associated scripts or transcripts when available should be evaluated in terms of their relative benefit in relation to cost to implement and support.

Enhanced batch metadata tools

During the development of HydraDAM, a small additional funding pool became available to WGBH that was used to implement a minimal batch ingest mechanism. The code supporting batch ingest has since been ported to other Hydra applications, notably for Tufts University and Case Western Reserve University. In both cases, the batch functionality has been extended and enhance. An obvious next step for HydraDAM would be to re-incorporate these external enhancements and tailor them to work with media-based workflows.

Enhanced content and metadata exchange tools

As mentioned above, HydraDAM currently leverages the Sufia user interface very heavily. That interface was designed to support web browser based interaction with the repository by human end-users. Due to the potential richness of the content which might be stored in a HydraDAM based repository, it would be a logical

extension to incorporate a machine-readable API into the system. Potential easy wins here would be the implementation of Blacklight based micro-data (machine readable descriptive metadata tags) within the existing item level views alongside an implementation of the Resource Sync framework to permit higher level synchronization of metadata between HydraDAM and other Resource Sync compliant systems.

PROJECT CONCLUSIONS

Given the necessary constraints on the project scope based on available funding and delivery deadlines, the project was exceptionally successful. The success of the project is demonstrated by the multiple ways in which the project has benefitted WGBH and the larger Hydra Community:

- Sufia, along with a number of other Hydra projects have been able to incorporate transcoding and media playback capabilities developed specifically to support the WGBH project.
- Discussions were begun about how to integrate the content distribution and streaming capabilities available in Northwestern and Indiana University's joint Avalon Media System project with the metadata management and preservation capabilities of the HydraDAM system. HydraDAM would be the preservation repository that can feed content into Avalon for access. A new NEH grant will support this work.
- The support documentation generated as part of the project has evolved into an independent work stream being evolved and maintained independently within the Hydra community.
- WGBH is contributing developer time to an agile development sprint with Penn State to update Sufia to take advantage of new functionality with Fedora 4. The upgrades will be useable for HydraDAM.
- Finally, WGBH is currently moving forward with a production implementation of the system intended to replace their current archival management system.

The current version of the application can be found at
https://github.com/curationexperts/hydradam
The wiki hosting supporting documentation can be found at
https://github.com/curationexperts/hydradam
As this is a live application, we expect both the application and documentation to evolve as the project grows.

The project has already had significant impact within the Hydra community. With multiple projects in development to enhance and expand the system's capabilities beyond the scope identified within the initial NEH grant, we feel very comfortable that the project has fully met, and exceeded, these stated goals:

- Developing open materials, tools, and documentation beyond the scope of the project
- Creating a content agnostic system that is flexible enough to evolve over time as media technology changes
- Collaborating across different communities of practice to assemble an end-to-end digital asset management system that provides solutions and services supporting the public broadcast community

Further Observations/Lessons Learned

What did we learn about Open Source solutions? Well they are not free. They do take attention and care to build and evolve. However, you can build off the work of others and significantly cut down on the work you need to do. In addition you can collaborate with others in the community to evolve systems and add common shared features.

The Hydra community shows promise in growth and sustainability. For us, it was a little bumpy at first with the timeline to develop features not quite lining up with other institutions timeline to develop the same features, but as more features are built, it becomes easier to adopt the code and quicker to have a fully functioning system.

As with any Open Source project, it is never done. If WGBH wants to continue to use HydraDAM we will need continue to develop it and evolve it, updating gems and code to make it compatible to the newest technology and newest challenges of managing media files. It will be to our advantage to contribute to the community, stay involved, and continue to help keep it strong.

One lesson we learned is that if you plan to build and maintain a system like this having at least two developers on staff is helpful. Loss of staff can greatly affect progress. Most developers like working in teams, and if one leaves, the other has enough knowledge of the system to be able to continue while management looks for a replacement. One advantage of Hydra was that it is based on Ruby and Rails, which seems to have a fast learning curve for a new developer to get up to speed. In addition, because many commercial websites are using that language, more and more developers are learning it, broadening the pool of candidates. However, one is also competing with those commercial organizations for these candidates and therefore the salary level is higher. We, in the non-profit cultural heritage circle, should emphasize the goals of the organization and mission, and aim to draw more candidates interested in the work itself rather than just the salary potential. So far that has attracted dedicated developers who have embraced this vision.

Hiring developers on a contract basis is a viable option when looking for additional staff, either on a monthly basis or for short development cycles such as sprints on a specific project. This requires you to quantify tasks and outcomes to keep the project moving forward, especially if it concerns user interface matters as opposed to back-end support.

We were trying to answer the question of whether an open source solution like HydraDAM makes sense for small cultural institutions. Developing a solution from scratch may be too much for them to undertake. The technology changes and evolves so quickly that is it hard to keep up with migrations, new versions, and with the community overall. One observation in trying to launch a project like this and working with the community was that timelines for project solutions across different institutions are different. It takes patience, coordination, and shifting priorities to take advantage of solutions coming out of the community. A small organization with limited resources may or may not have that time.

Building off another product helped us leap forward in our project. Once more formal solution bundles are developed, which is currently happening with Avalon and Sufia, there will be readily available Hydra applications to download and install that have better documentation and are more suitable for smaller institutions with fewer tech resources to implement. Our project did illustrate this with successful implementations at two organizations with little to no Hydra knowledge. It also highlighted the need for thorough documentation and piecemeal installation instructions. More Hydra vendors and developers, and the ability to hire them on contract rather than staff, would help enormously in this regard.

In the end, we challenged the idea of needing a complicated HSM system to manage our volume of digital media files. Although HydraDAM has hooks that can be built to connect to an HSM system, we have chosen to simplify our workflow to utilize localized LT06 workstations to capture storage of our preservation files. In the past we relied on a networked HSM system and robotic library of LTO tapes. We found this caused problems moving the large files across the network, and is very expensive to maintain. We are now directly laying preservation files onto LTO06 tapes with a local workstation. Those tapes are then stored in our vault as any other physical archive asset. The data upon ingest is going to one central repository.

Our current work is developing the code to make the localized connections from HydraDAM to the location of the preservation files on the LT06 tapes. By default, HydraDAM is installed on a server than should have 30-300GB of available storage space, according to the documentation. This is a default configuration suitable for small institutions that may not have HSM or LTO storage.

We found the Hydra community to be very welcoming and helpful. Lots of work can get done in virtual spaces connecting over the network and by phone. But

participating in face-to-face meetings like Hydra Connect and meeting at other conferences like Open Repositories allows a familiarity and trust to build that is important when working in collaborative communities. The value of those in-person meetings should not be underestimated.

Regardless, our open source system is a "free puppy," and not a "free beer." It is not an easy, cheap solution, it needs attention and care in order to thrive and to fulfill the needs of media preservation. From our SETV feedback:

"Since the HydraDAM has been operational, we've created a working group to test the system. Our first phase is to take a project that was recently completed and populate HydraDAM with all the metadata and video. The first couple of meetings resulted in recommendations to add more storage, increase the video file size limit of 200 Meg and add a least three fields or rename existing to show "Project #" or "Episode" and possible Project Name along with our in house "media ID".

Phase 2 will be an active weekly program that starts back for a new season in November. It's called "Palmetto Scene". All clips and related docs for the project will be ingested as they come in or created and HydraDAM will be the pre-production tool. We're very cognizant of possible workflow changes for this show and are working proactively with the production team.

So far, our questions and processes have been worked out with ease through group interaction and a few emails. No stoppers so far. We're encouraged."

We would like to thank NEH for their support of this project and we hope it has contributed to the community in reaching a solution for a very challenging issue.

GLOSSARY

Avalon (http://avalonmediasystem.org/)
Avalon is an open source Hydra solution bundle developed by Indiana University and Northwestern University to provide online streaming access to digital audio and video.

Blacklight (http://projectblacklight.org)
Blacklight is an open source Ruby on Rails gem that provides a discovery interface for the Solr search engine.

Fedora (http://www.fedora-commons.org)
Fedora Commons is repository software enabling long-term access to digital

resources. A key feature of Fedora is its flexibility in support all types of digital content.

FFMEG (http://www.ffmpeg.org)
FFMPEG is a complete, cross-platform solution to record, convert and stream audio and video.

FITS (http://projects.iq.harvard.edu/fits)
The File Information Tool Set (FITS) identifies, validates and extracts technical metadata for a wide range of file formats.

Free Puppy/Free Beer (https://www.gnu.org/philosophy/free-software-for-freedom.html)
Our analogy to describe the work required to maintain a "free" open source application. Much like if someone were to give you a free puppy, it's not really free because you would still have to buy it food, take it to the veterinarian and invest the time to raise it correctly. This is similar to open source software because it takes skilled staff to operate and maintain any no-cost application you implement.

GNU LPGL License (https://gnu.org/licenses/gpl.html)
This a common license attributed to free, open sourced software on the Internet.

Hydra (http://projecthydra.org)
Hydra is a digital repository framework integrating Fedora, Solr, and Blacklight into a unified application stack. The hydra framework integrates these and other tools to enable the rapid development of highly-functional user-facing repository solutions.

PBCore (http://www.pbcore.org)
PBCore is a metadata standard developed and used by the public broadcasting community in the United States.

RAID (http://en.wikipedia.org/wiki/RAID)
RAID stands for redundant array of independent disks and refers to formatted computer storage (like hard drives) that are combined through software together to virtually create a larger, or more redundant single storage unit. One example, two seperate 1TB hard drives can be RAID written to be useable to a computer as a single 2TB hard drive.

RDF (http://www.w3.org/RDF/)
RDF is a standard model for data interchange on the Web. RDF allows structured and semi-structured data to be mixed, exposed, and shared across different applications and institutions.

SAN (http://en.wikipedia.org/wiki/Storage_area_network)
SAN stands for storage area network and a SAN system often times is used to provide larger storage and access across a networked connection.

Schema.org microdata (https://schema.org, see also http://en.wikipedia.org/wiki/Microdata_(HTML))
Structured data markup schema supported by major search engines to provide richer search results. This markup also enables new tools and applications to use data embedded in human readable web pages.

ScholarSphere (https://scholarsphere.psu.edu)
ScholarSphere is a secure repository service based on the Hydra framework enabling the Pennsylvania State University community to share its research and scholarly work with a worldwide audience.

Solr (http://lucene.apache.org/solr)
Solr is a popular open source enterprise search platform providing full-text search, faceted search, and near real-time indexing.

Sprint (http://en.wikipedia.org/wiki/Scrum_(software_development)#Sprint)
Software development term used to describe a focused, planned period of time when the development team is assigned work to complete.

SSD (http://en.wikipedia.org/wiki/Solid-state_drive)
SSD stands for solid-state drive (or disk) and is a type of computer storage, often times found as hard drives. SSDs have no mechanical, spinning disk like regular hard drives. A USB thumb drive is also an example of an SSD.

Sufia (https://github.com/projecthydra/sufia)
Sufia is a Rails engine for creating a self-deposit institutional repository. A web application for ingest, curation, search, and display of digital assets. Powered by Hydra technologies: Rails, Hydra-head, Blacklight, Solr, Fedora Commons, etc.

TRAC (http://www.crl.edu/archiving-preservation/digital-archives/certification-and-assessment-digital-repositories)
The Trustworthy Repositories Audit and Certification (TRAC) checklist defines a set of metrics to be used in assessing long-term preservation repositories, based on the OAIS (Open Archival Information System) model. TRAC has informed development of the ISO 16363 standard for audit and certification of trustworthy repositories.

Features / Requirements

Core Functionality

- Easy retrieval of web-displayable versions of content (when available)
- Access controls use Hydra rightsMetadata (not XACML)
- Ingest: (asset first) Single-file upload via browser
- Well-documented API enables Ingest via scripts
- Support ingest models for at least 3-4 common file formats as specified by WGBH
- Easily implemented on common off-the-shelf hardware while allowing for scalability to petabytes of material

Audio/Video

- Specifications and recommendations for ingest models for at least 3-4 common file formats.
- Transcoding and technical metadata extraction on ingest
- Creation of a proxy and a thumbnail for viewing access for at least 3-4 common digital media (video) file formats
- Front-end access for at least 3-4 common digital media file formats
- Retrieval of master asset files for at least 3-4 common digital media file formats
- The system is bundled with a TRANSCODING SERVICE that can perform transcoding tasks and extract metadata & thumbnails

Large Files

- System provides downloads of large files via temporary opaque FTP urls
- Ingest of Files via "Watch Folder"/"Drop Folder" per user

Management

- PBCore will be utilized as the metadata structure.
- Users can edit a page with PBCore metadata fields (more than 9 fields, less than 100)

Preservation

- Verifiable ingest (checksums on ingest)
- Periodic file checksum
- Logging and reporting of Access (views) & Exports - for TRAC compliance
- Support for integrating HSM Solutions, included abstraction layer allowing alternate future HSM implementations

Quality Assurance

- Implement and test the system, and will provide feedback to improve both the software and documentation
- Test the ease of implementation of the software at 2 partner public media organizations.

NOTE: continuous testing and feedback are part of the agile development cycle; therefore, general QA costs are integrated into feature development costs.

Documentation

- Documentation for easy replication and implementation.
- Document methods for handling local filesystems, HSM, and Cloud Storage with Abstraction Layer
- Develop documentation needed for implementation of the digital repository bundle.
- Develop written documentation needed for implementation and local customization of the Hydra “head”.
- Document the skills set necessary to implement the system
- The storage location of files on the storage medium will be documented
- Set up development and staging environments

Out of Scope

- Complete access and management interfaces to support discovery and metadata cataloging/enhancement
- Content workflows for accession/review/publish
- Tools to monitor the performance and use of the system (dashboards, analytics, systems monitoring)
- Handling of additional Media Types & file formats